(12) UK Patent Application (19) GB (11) 2 271 260 (13) A

(43) Date of A Publication 06.04.1994

(21) Application No 9220792.7

(22) Date of Filing 02.10.1992

(71) Applicant(s)
Canon Kabushiki Kaisha

(Incorporated in Japan)

30-2 Shimomaruko 3-chome, OHTA-KU, Tokyo 146, Japan

(72) Inventor(s)
Adam Michael Billyard

(74) Agent and/or Address for Service
Beresford & Co
2-5 Warwick Court, High Holborn, LONDON, WC1R 5DJ, United Kingdom

(51) INT CL^5
G06F 15/72

(52) UK CL (Edition M)
H4T TBBA

(56) Documents Cited
GB 2194715 A GB 2194656 A EP 0152741 A2

(58) Field of Search
UK CL (Edition K) H4T TBBA TBEA TCGA TCGD TCGG TCGK TCGX TCJA TCXX
INT CL^5 G06F

(54) Processing image data

(57) In apparatus for processing image data representing multi-dimensional objects, data is stored defining the position of a light source, a viewing position, an arrangement of polygons defining an object and a local transform for transforming said object into viewable 3-dimensional space.

Processing means defines the extent of points for an object in each dimension to establish a cuboid bounding box 121, 122, 123 for the object and performs said local transform on said box and determines the extent to which said points (e.g. for boxes 32a-c are transformed into viewing space 81.

Thus, an evaluation can be made as to where the object will lie in viewing space before polygons are transformed. Culling may then be performed at the object level, rather than at the polygon level.

FIG. 12

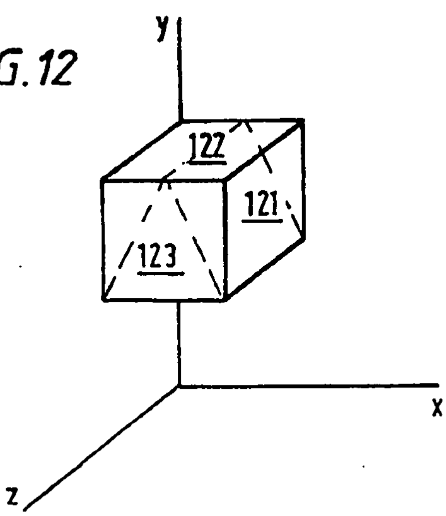
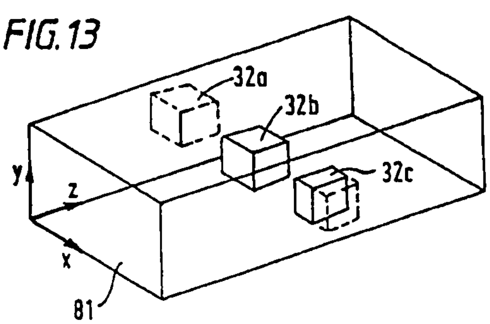


FIG. 13



GB 2 271 260 A

FIG. 1

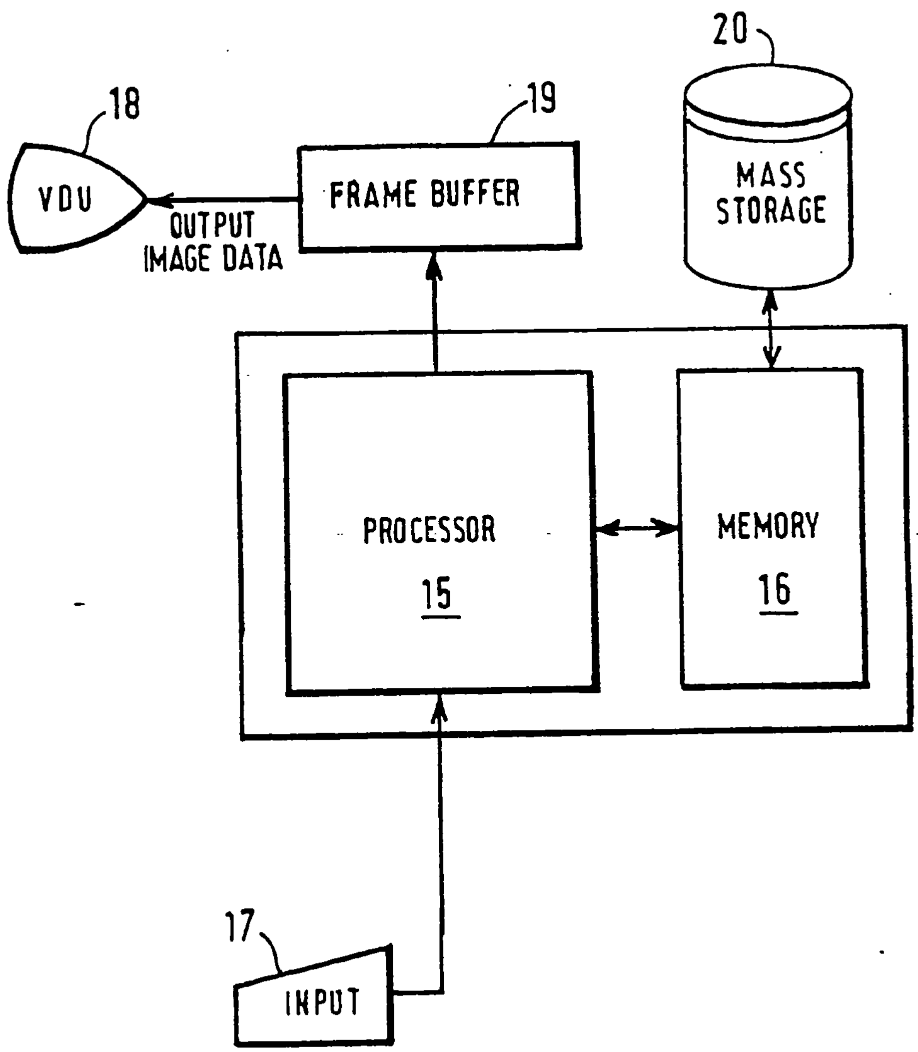
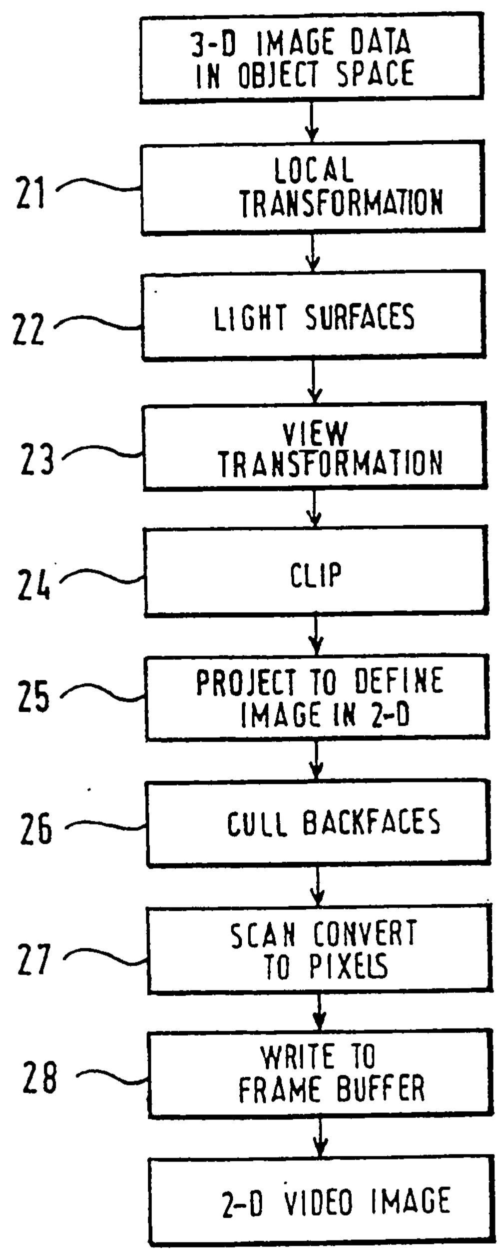

18
VDU
19
FRAME BUFFER
OUTPUT
IMAGE DATA
20
MASS
STORAGE
PROCESSOR
15
MEMORY
16
17
INPUT

FIG. 2


3-D IMAGE DATA IN OBJECT SPACE
21
LOCAL TRANSFORMATION
22
LIGHT SURFACES
23
VIEW TRANSFORMATION
24
CLIP
25
PROJECT TO DEFINE IMAGE IN 2-D
26
CULL BACKFACES
27
SCAN CONVERT TO PIXELS
28
WRITE TO FRAME BUFFER
2-D VIDEO IMAGE

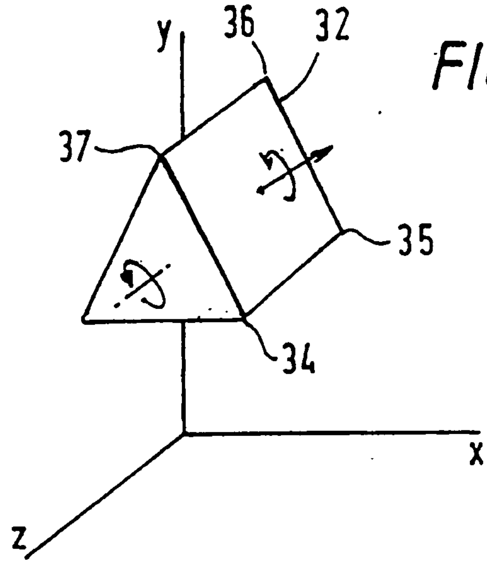


FIG.3

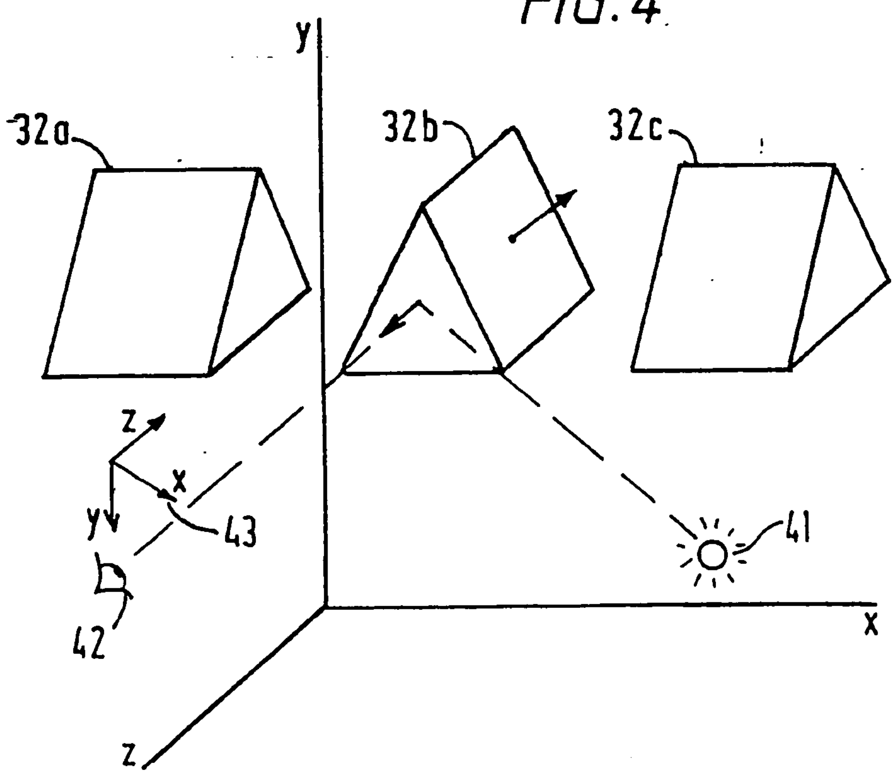
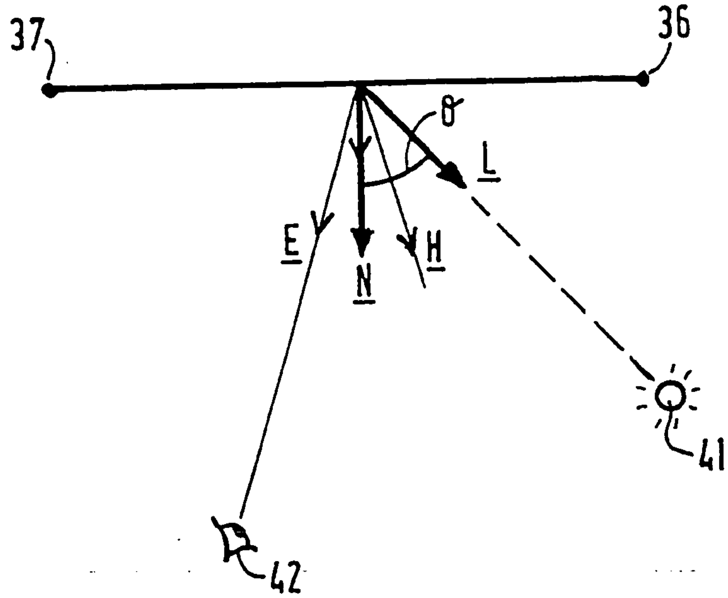


FIG.4

FIG.5



$$I(a) = B(a) \,.\, K(a)$$

$$I(d) = I(l) \,.\, K(d)\, \underline{N} \,.\, \underline{L}$$

$$I(s) = I(l) \,.\, K(s)(\underline{H} \,.\, \underline{N})^n$$

$$I(RGB) = I(a) + I(d) + I(s)$$

FIG.6

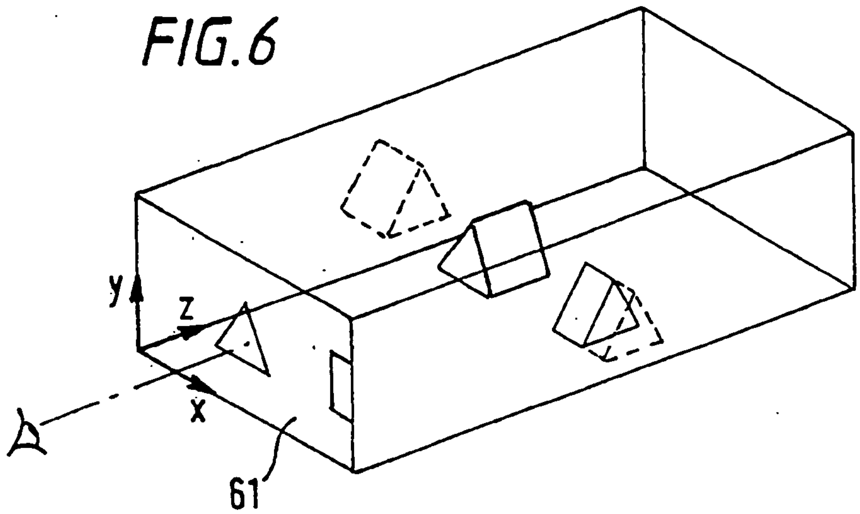


FIG.7

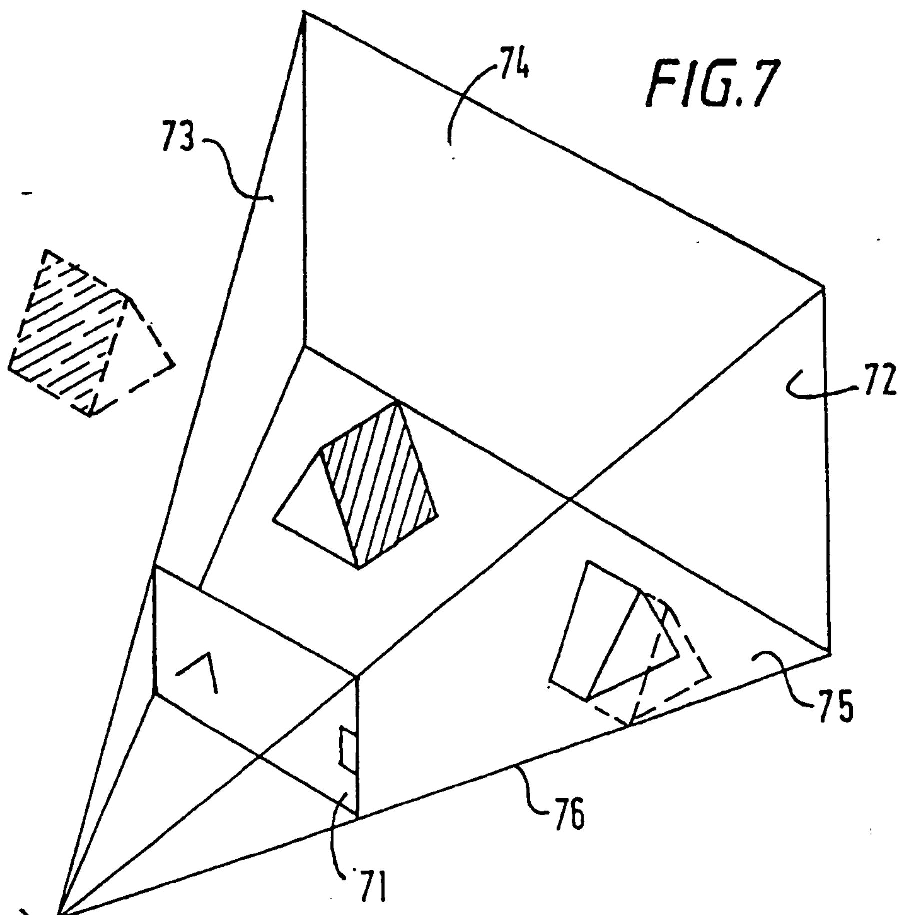
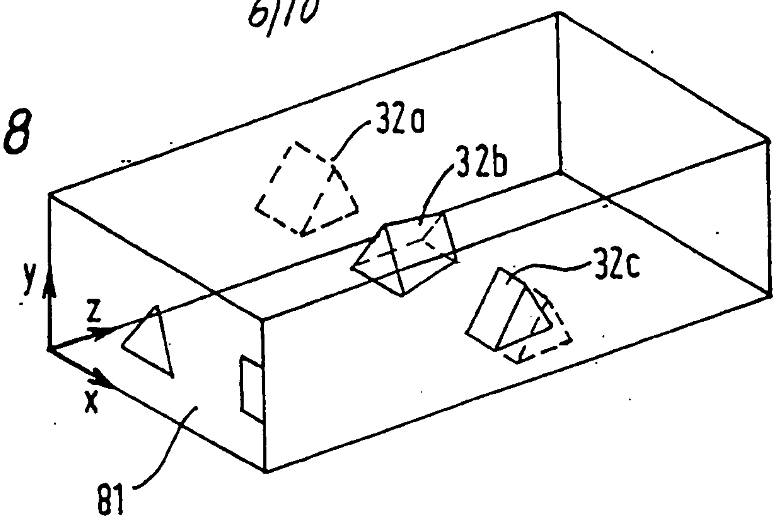


FIG.8



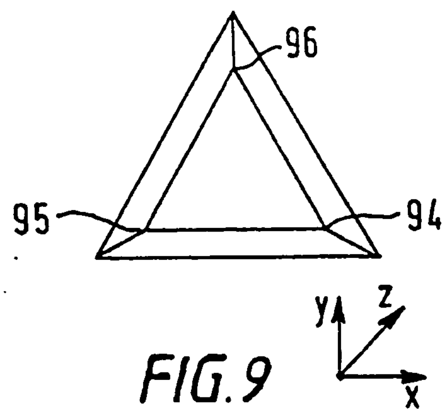


FIG.9

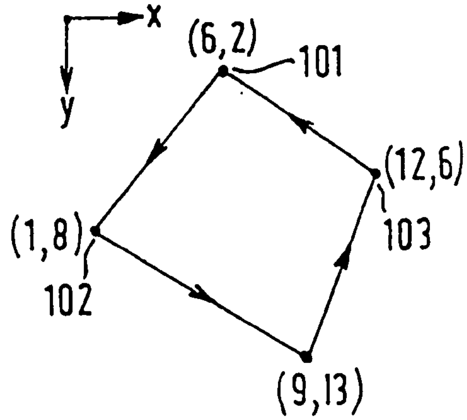


FIG.10

FIG. 11

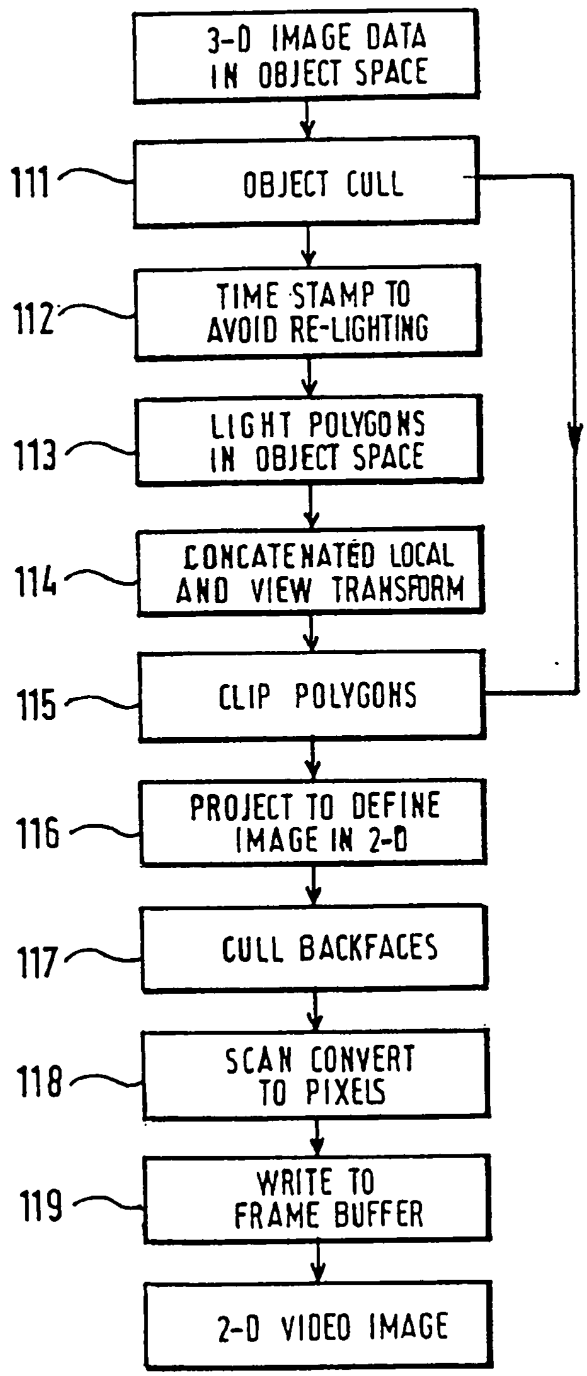

3-D IMAGE DATA IN OBJECT SPACE
111
OBJECT CULL
112
TIME STAMP TO AVOID RE-LIGHTING
113
LIGHT POLYGONS IN OBJECT SPACE
114
CONCATENATED LOCAL AND VIEW TRANSFORM
115
CLIP POLYGONS
116
PROJECT TO DEFINE IMAGE IN 2-D
117
CULL BACKFACES
118
SCAN CONVERT TO PIXELS
119
WRITE TO FRAME BUFFER
2-D VIDEO IMAGE

FIG. 12

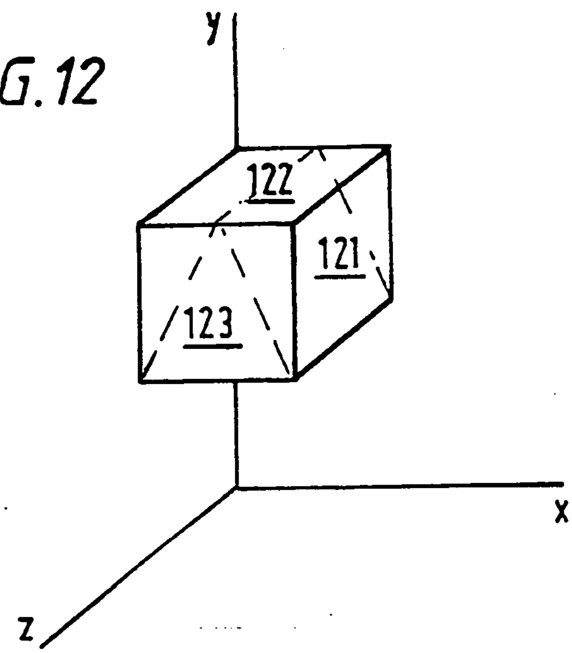


FIG. 13

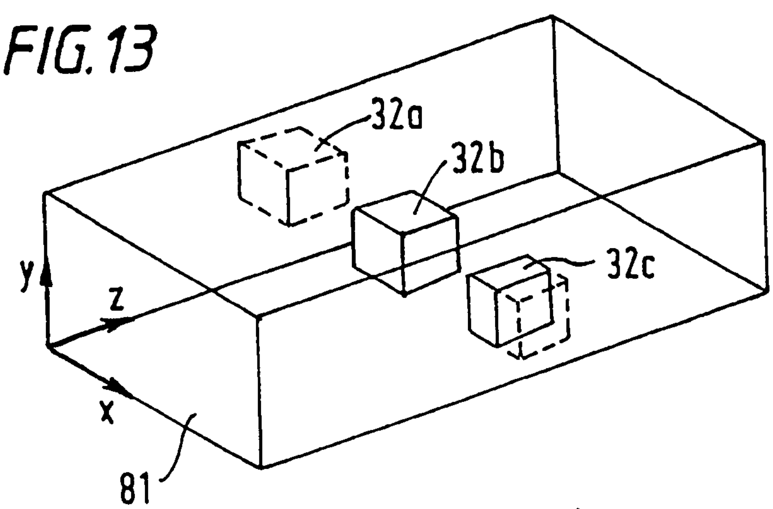


FIG. 14

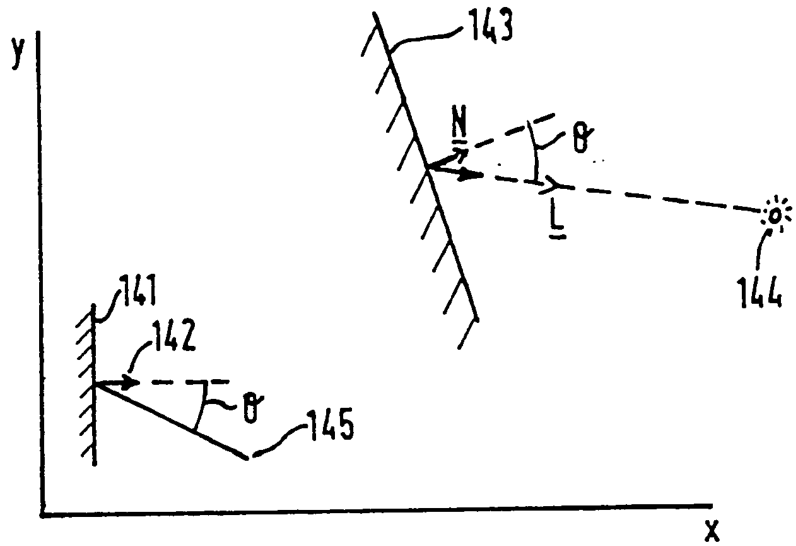


FIG. 15

$$I(d) = I(l) \; . \; K(d) \; \underline{N} \, . \, \underline{L}$$

$$I(s) = I(l) \; . \; K(s) \; . \; (\underline{N} \, . \, \underline{L})^{n}$$

FIG. 16

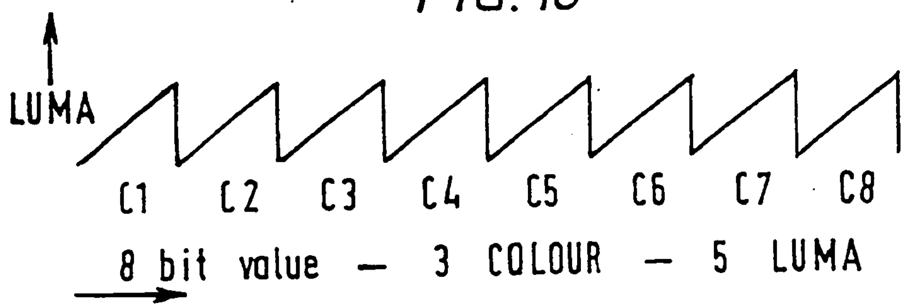


FIG. 17

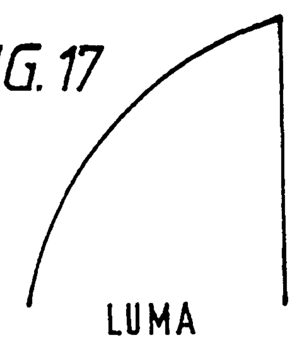
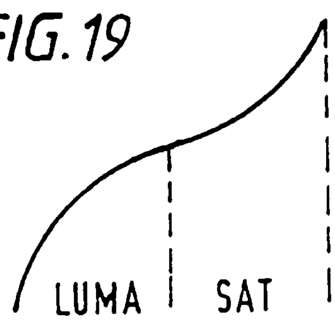


FIG. 18

FIG. 19



2271260

PROCESSING IMAGE DATA

FIELD OF THE INVENTION

The present invention relates to apparatus and methods for processing image data.

In particular, the present invention relates to apparatus and methods for processing image data for use in interactive 3-dimensional graphics environments. The present invention also relates to apparatus and methods for generating 2-dimensional representations of 3-dimensional objects.

BACKGROUND OF THE INVENTION

System are known which are capable of synthesizing 2-dimensional images in response to data defining elements within a 3-dimensional space. The final 2-dimensional result may consist of a very large number of coloured picture elements (pixels) which may be viewed on a monitor or printed onto an image carrying medium.

In interactive systems arranged to generate data representing a 3-dimensional space, objects appear to move within the 3-dimensional space in response to input commands. Thus, in such systems a machine is required to render a 2-dimensional image from data representing a 3-dimensional space and, in addition,

the machine is also required to perform this operation repeatedly as the position and/or orientation of objects, light sources and the view point change in response to input commands. Typically, in an interactive environment, the machine is required to produce output images at a rate of between five to fifteen per second. In more highly powered environments, output images are produced at video rate (60 frames per second) and the machine is said to operate in real time.

In known machines, a detailed example of which will be described later, a significant amount of hardware is required if acceptable images are to be produced at an interactive rate. Thus, computational demands have placed significant constraints on 3-dimensional interactivity, thereby limiting the extent to which interactive 3-dimensional graphics may be employed.

It is an object of the present invention to provide apparatus and methods for processing image data, which require reduced computational overheads.

It is a further object of the present invention to provide apparatus and methods capable of producing acceptable images at an interactive rate but with lower computational demands.

In known 3-dimensional graphics processing systems, a great deal of data is processed unnecessarily, in that it does not provide a contribution to the final output image, because such systems are inflexible and are not capable of selective processing.

It is an object of the present invention to provide an improved system. In particular, it is an object of the present invention to provide an improved system with a level of selectivity, thereby enabling it to reject data which does not require further processing.

SUMMARY OF THE INVENTION

According to an aspect of the present invention, there is provided apparatus for processing image data representing multi-dimensional objects, some apparatus including processing means and data storage means, wherein said storage means stores data defining; the position of a light source, a viewing position, an arrangement of polygons defining an object and a local transform for transforming said object into viewable 3-dimensional space, characterised by said processing means including means for: defining the extent of points for an object in each dimension; performing said local transform on said points and determining

the extent to which said points are transformed into viewing space.

Thus, the present invention provides a test for determining whether an object will be present in viewing space. On the basis of said test, decisions may be taken concerning the extent of further processing.

The invention will now be described by way of example only, with reference to the accompanying drawings, in which:

Figure 1 illustrates an environment for implementing embodiments of the present inventions;

Figure 2 illustrates a conventional approach to interactive 3-dimensional graphics;

Figure 3 represents an object fabricated from polygons in its own object space;

Figure 4 represents an object repeatedly transformed into modelling space, including a light source and a viewing position;

Figure 5 illustrates techniques for calculating lighting characteristics from light source data, viewing data and surface parameters;

Figure 6 shows a viewing space for an isometric projection;

Figure 7 shows a frostrom providing the first

stage of a viewing transformation for perspective views;

Figure 8 shows a viewing space derived from the frostrom of Figure 7 for generating perspective views;

Figure 9 illustrates a projection of polygons into 2-dimensions;

Figure 10 illustrates a projected polygon for the purpose of scan conversion;

Figure 11 shows an embodiment of the present invention, facilitating interactive 3-dimensional graphics production;

Figure 12 illustrates a bounding box defining extents of an object defined in its own object space;

Figure 13 illustrates the transformation of the bounding box shown in Figure 12 into viewing space;

Figure 14 illustrates the calculation of lighting characteristics in accordance with an embodiment of the present invention;

Figure 15 illustrates algorithms used for calculating diffuse and specular reflections in accordance with an embodiment of the present invention;

Figure 16 illustrates a colour look up table for converting eight bit colour values into full colour

values for display on a visual display unit, including a plurality of colour ramps;

Figure 17 illustrates an alternative colour ramp with gamma correction for luminance values;

Figure 18 illustrates an alternatively shaped colour ramp with non-linear saturation values; and

Figure 19 illustrates an alternative colour ramp combining non-linear luminance and saturation values.

A CONVENTIONAL INTERACTIVE GRAPHICS SYSTEM

An environment for synthesizing 3-dimensional data, viewing 2-dimensional data derived from said 3-dimensional data and interactively modifying said 3-dimensional data is shown in Figure 1. A processor 15 is arranged to write data to and read data from a memory device 16. Data stored in the memory device 16 may define 3-dimensional image data, 2-dimensional image data, instructions to the processor 15 and other types of data which may or may not be relevant to interactive 3-dimensional graphics.

In addition, the processor 15 receives input data from input devices 17, said devices consisting of a manually operable keyboard and a position sensitive device, such as a mouse, a tracker-ball or a digitising tablet with a stylus etc.

2-dimensional images are displayed on a visual display unit 18, which receives output image data at video rate, by raster scanning a frame buffer 19. The visual display unit 18 may have a definition of 1,000 lines with 1,000 pixels on each line, requiring frame buffer 19 to include 1,000,000 pixel locations.

For the bulk transfer of program data and image data, a mass storage device 20 is provided, such as a hard magnetic disk, optical disk or tape drive etc.

The combination of processor 15, memory device 16, storage device 20, frame buffer 19 and the visual display unit 18 may be commercially available as a complete system. For example, the configuration may consist of a Sparc workstation, supplied by Sun Microsystems Inc. of the United States.

In response to input commands from device 17, image data is written to the memory device 16 defining positions in a virtual 3-dimensional space. A plurality of 3-dimensional spaces are provided and transformations are made between them, the nature of which will be detailed later. In response to control data from the memory device 16, the processor 15 is arranged to generate a 2-dimensional view from data representing 3-dimensional space. 2-dimensional views of this type are built up in the memory 16 and

supplied to the frame buffer at a rate of between five to fifteen frames per second. The frame buffer 19 is then read at video rate, such that data supplied to the buffer may be read more than once, so as to maintain a stable image.

A conventional system for interactive 3-dimensional graphics is shown in Figure 2. 3-dimensional objects are defined in terms of a plurality of polygons. Each polygon is in turn defined as a plurality of connected points, referred to herein as vertices. Thus, objects are created by joining polygons, having coincident vertices along an edge common to both of the polygons.

A 3-dimensional object is defined within its own object space, that is to say, the vertices of polygons are positioned with respect to the object's own coordinate axes.

An object space of this type is shown in Figure 3. In accordance with convention, the origin of the X, Y and Z axes is positioned at the back bottom left and the vertices of polygons are given x, y and z Cartisian coordinates with reference to this origin.

In Figure 3, a simple object is shown, made up of five polygons. Data is stored in memory defining the position of each vertex of each polygon. For example,

polygon 32 has four vertices 34, 35, 36 and 37, the coordinates of which are listed in a table. By convention, the vertices of a polygon are defined in an order obtained by tranversing around the polygon in an anticlockwise direction.

In addition to this positional information, additional information relating to the polygon is also stored. This includes a definition of a unit normal vector 38, a vector extending from the centre of the polygon in a direction perpendicular, i.e. normal to the plane of the polygon, away from its front face, and extending a length equal to unity within the coordinate reference frame. Furthermore, data is also stored relating to the nature of the surface of the polygon and in particular parameters are stored defining how the polygon will react with modelled light directed thereon.

Within a modelling space, in addition to objects, light sources are also provided and a viewing position is selected. Light sources model light falling upon the polygons, which in turn have lighting characteristics, that is to say, the polygons have colour and brightness responsive to the amount of light falling upon them and the nature of their surface.

Thus, within the table defining the polygons of the object shown in Figure 3, data is stored relating to a coefficient of ambient light K(a), a coefficient of diffuse light K(d) and a coefficient of specular light K(s).

The coeffient of ambient light effectively defines how bright a polygon is when no local light sources are provided. The coefficient of diffuse light is a parameter which defines the polygon's characteristic relating to the diffusion of light in response to a local light source. Similarly, the K(s) defines characteristics relating to specular highlights. Thus, the presence of specular highlights effectively represents how shiny a polygon is and improves realism.

Returning to the system detailed in Figure 2, the 3-dimensional image data in object space, as shown in Figure 3, undergoes a local transformation which transforms objects into modelling space, at step 21. At step 22, the polygon surfaces are lit by processing data relating to the position of light sources in modelling space and the lighting parameters of each polygon. Thereafter, at step 23, the modelling space is transformed into a viewing space, the geometry of which differs depending upon whether an

isometric view or a perspective view is required.

The view transformation will identify a particular field of view, which will usually cover less than the whole modelling space. Therefore, at step 24 a clipping process is performed to remove polygons which fall outside the field of view.

Up until this stage, data processed by the processor and read from the memory defines 3-dimensional coordinate locations. At step 25 the projected view is analysed to define an image in 2-dimensions, thus, in isometric views, the z dimension is rejected and in perspective views the x and y dimensions are scaled, so as to take account of the rejected z dimension.

After projecting the image into 2-dimensions at step 25, it is necessary to identify the difference between front faces of polygons and back faces of polygons so that the back faces may be removed, given that they cannot actually be seen because they are obscured by a front face. Thus, at this stage, vertices are defined in 2-dimensions identifying the front faces of visible polygons.

At step 27, the 2-dimensional data relating to polygon position is scan converted to produce coloured pixels, taking into account data defining the colour

of each polygon, as previously calculated.

At step 28, the pixels are written to the frame buffer on a polygon-by-polygon basis, thereby building up a complete 2-dimensional image. It should be noted that images must be written to the frame buffer at a rate of between ten to fifteen frames per second in order that a viewer will feel that the system is responding in an interactive way to input commands.

A representation of modelling space is shown in Figure 4. World coordinate axes X, Y and Z have a similar orientation to the axes provided in object space, as shown in Figure 3, but may be scaled differently. For example, it may be appropriate, under default conditions, to define objects as being relatively large in object space and relatively smaller in modelling space, so that a plurality of objects may be assembled in modelling space.

In the example of modelling space illustrated in Figure 4, the object illustrated in its own object space of Figure 3, has been instantiated into the modelling space three times. Furthermore, some of the objects have been rotated, so that some normals 38 extend from the centre of polygon 32 arc pointing in the positive Z direction, whereas, in object space, they were pointing in the positive X direction.

Transformations from an object space, such as that illustrated in Figure 3, to a modelling space, such as that illustrated in Figure 4, are effected by matrix manipulation. Thus, the coordinate positions of, say, points 34, 35, 36 and 37 are multiplied by a transform matrix which replaces coordinates within the object space of Figure 3 with coordinates of modelling space as in Figure 4. Three instantiations of the object are placed into modelling space by using three different matrix transforms. Furthemore, additional objects may be placed into modelling space by effecting transformations upon other objects defined in their own object space.

Light sources may be introduced into the modelling space which, in addition to having position, also have orientation. Furthermore, parallel light may be introduced into the modelling space, representing the effect of light eminating from an infinitely displaced source, similar to light produced by the sun. The effect of each light source is considered in turn and lighting effects for each source are added together on a polygon-by-polygon basis.

In addition to the position of light sources, a viewing position 42 is also selected, illustrated by an eye symbol. The viewing position may,

alternatively, be represented by a set of axes X, Y, Z 43, given that a transformation from modelling space as shown in Figure 4, consists of transforming objects into a new space,which has a different set of axes. In particular, whereas, in modelling space, the origin is at the back bottom left, in viewing space the origin is at the front bottom left. Thus, the Z direction represents distances measured away from the viewing position.

As shown in Figure 2, lighting calculations are made at step 22 before the viewing transformation is effected at step 23.

Lighting calculations for polygon 32b will be considered, with reference to Figure 5. The lighting calculation produces an illumination factor I which, in a colour system, consists of colour components RGB.

The value I(RGB) representing the total illumination factor for the polygon, consists of an illumination factor I(a) from ambient lighting, a component I(d) from diffuse lighting from a light source and a component I(s) for specular lighting from said light source. I(a) is derived from the background ambient lighting B(a) and is obtained by multiplying this value by the ambient coefficient K(a), as shown

in Figure 5.

I(d) and I(s) are dependent upon the position and intensity of the light source 41 and, in accordance with the Phong model, I(s) is dependent upon the position 42 of the observer. The Phong model is preferred in systems of this type, although other models are known.

In the model, the diffuse component I(d) is derived from the cosine of the angle theta between a vector pointing towards the light source 41 and a vector normal to the plane. Its value is calculated by the dot product of the unit normal vector $\underline{N}$ and a unit vector $\underline{L}$ pointing in the direction of the light source 41. Thus, the component I(d) is obtained by multiplying a value identifying the intensity of the light emitted by the light source 41 I(l) by the dot product of vectors $\underline{N}$ and $\underline{L}$ and by the coefficient K(d).

In accordance with the Phong model, the intensity of specular highlights is dependent upon the position of the viewer 42. A unit vector $\underline{H}$ is calculated which is half way between the unit vector $\underline{L}$ and a unit vector $\underline{E}$, directed towards the position of the viewer 42. The dot product is then formed between the unit normal vector $\underline{N}$ and the unit half way vector

H . The model is arranged such that the specular highlight is large when vectors E and L form similar angles to the plane 36, 37, thus resulting in vectors N and H being coincident. Moving away from this position results in the intensity of the specular highlight decreasing rapidly. To represent this in the model, the dot product of the H and N is raised to a power, typically lying within the range 15 to 20 or higher. This value is then multiplied by the coefficient K(s) and the intensity value I(1).

As previously stated, the total intensity value for the polygon in RGB space is calculated by summing I(a) with I(d) and I(s).

Raising the dot product of H and L to a power is a computationally expensive process, therefore the inclusion of means for calculating specular high lights is a significant overhead and in many systems is not included. However, specular highlights add significantly to the realism of the object and allow distinctions to be made between shiny and matt surfaces.

The lighting calculations illustrated in Figure 5 are repeated for each polygon at step 22 of Figure 2. As previously stated, at view 23 a view transformation is made, taking account of the view position 42. It

should be noted that, in response to interactive input, the view position 42 may be interactively modified, therefore, in the same way in which local transformations and the lighting of surfaces are performed at the interactive rate, view transformation is also performed at the interactive rate.

Most view positions result in only part of the modelling space being viewed, therefore, essentially, a viewing transformation consists of defining the viewable space and shifting the axes.

A representation of a transformation of the modelling space, shown in Figure 4, into isometric viewing space is shown in Figure 6. As can be seen, each vertex is measured from the bottom left corner of a front viewing plane 61 and a viewer is positioned normal to the viewing plane in a minus Z direction.

A perspective view transformation may be considered in two stages. As shown in Figure 7, the viewable portion of the modelling space is considered to be a frustrum, bounded by a front viewing plane 71, a back viewing plane 72 and sloping side planes 73, 74, 75 and 76. In order for the objects positioned within the frustrum to be maintained in their original shape, the space itself is effectively non-Cartesian, therefore further transformation is required, such

that the viewing space itself can be represented in Cartesian coordinates. This results in a further transformation of the type illustrated in Figure 8. Herein, the frustrum itself has been transformed into a parallel pipe, by transformation under which the extent of negative enlargement increases along the Z axes. This results in negative enlargements being made to the objects within the space, such that edges that were parallel taper towards a vanishing point as they move in the positive Z direction. Thus, data generated under an isometric transformation, of the type illustrated in Figure 6, may now be processed in a similar manner to data generated under a perspective transformation of the type shown in Figure 8.

Referring back to Figure 2, after the 3-dimensional view transformation has been made, it is necessary to form a clipping operation at step 24 such that polygons which are not included within the viewable space are rejected and do not undergo further processing.

Referring to Figure 8, it can be seen that polygons will lie outside the viewable space. These polygons are identified by comparing their x, y and z coordinates with the coordinates of the viewable space, resulting in a list of polygons which will

require further processing.

Referring to Figure 2, the first stage of this further processing is to obtain a 2-dimensional projection of the 3-dimensional data by considering the image that would be seen at the front bounding plane 61/81.

Considering the central object shown in Figure 8, four polygons are visible from the viewing position as shown in Figure 9.

Vertices 94, 95 and 96 represent the vertices of the back face of the object which, as shown in Figure 8, has been reduced in size due to the perspective view transformation.

In addition, to transforming the vertices under the local and view transformations, the unit normal vectors are also transformed. Thus, the front polygon has a unit normal vector extending in the negative Z direction of the view coordinates, while the back face has a normal vector extending in the positive Z direction. Thus, at step 26 back face culling is performed by rejecting all polygons which have a unit normal vector with a positive Z component.

It will be appreciated, therefore, that the unit normal vectors are used at both step 22 when calculating lighting and at step 26 when back face

culling. Consequently, the unit vectors are transformed under the local transformation at step 21 and under the view transformation at step 23. Furthermore, these transformations will often produce normal vectors which are no longer of unit length. Consequently, it is necessary to re-normalise these vectors so that, in accordance with the coordinate system being used, they are re-established as normal vectors of unit length. Again, this is computationally expensive, given that square roots must be calculated for each polygon in order to calculate the x, y and z components of the transformed vector, having overall unit length.

Referring to Figure 2, after the back faces have been culled, step 27 consists of scan converting the polygons to produce pixel values.

An irregular polygon is shown in Figure 10 which has been projected into a 2-dimensional XY plane. The polygon has not been culled because its front face is showing, therefore descending the list of data coordinates results in anticlockwise traversing around the polygon.

The XY coordinates in the 2-dimensional projected plane define the positions of pixels within the frame buffer. Thus, the x and y coordinate positions

typically lie over a range of 0 to 1,000 in both the X and Y dimensions, from an origin at the top left corner.

Scan conversion is initiated by identifying which vertex of the polygon has the smallest y value, in this case, vertex 101. Thus, pixels within the polygon are filled-in starting from this point by calculating the gradients to vertices 102 and 103, which are located as positions (1, 8) and (12, 6) respectively. Thus, as known, it is possible to calculate which pixels within the polygon are to be painted with the polygon colour. This data is then written to the frame buffer at step 28.

Before data is written to the frame buffer, it is conventionally necessary to perform data truncation, given that the number of bits available within the frame buffer is limited, typically to eight bits per pixel location. Thus, a degree of colour aliasing must be tolerated, the extent of which being dependent upon the type of output device being used.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The processes previously described are successfully employed in interactive 3-dimensional graphics systems, in which 3-dimensional image data is modified in response to input commands and ouput

2-dimensional image data is generated at a rate sufficient to provide feedback to the operator, thereby completing the interactive environment.

The overiding problem with interactive 3-dimensional graphic systems is that of handling very large amounts of data. Each of the processes previously described can be implemented without difficulty and the computational requirements for performing any of the operations once is very modest. However, problems occur when the processing must be repeated many times.

Such a situation lends itself to processing within a hardware pipeline, in which a modest amount of processing is performed at each stage in the pipeline, whereafter the data is clocked to the next stage, resulting in a significant amount of parallel processing. Thus, the system shown in Figure 2 may be implemented as a hardware pipeline in which each process 21 to 28 is performed by its own dedicated hardware elements.

The amount of data being processed tends to increase dramatically as it moves along the stages, particularly given that data which represents the positions of vertices defining a surface must be processed to form data representing the colour of all

pixels within that surface. Thus, very large amounts of data are generated during step 28, during which pixel data is written to the frame buffer.

Hardware savings can be made by reducing, wherever possible, the level of data movement and data processing. It should be noted that, when a modest amount of processing is being performed on a large amount of data, even the modest process of moving data from one location to another results in a significant overhead, if all of the data under consideration must be moved in this way.

By implementing economies, the object is to reduce the number of physical processors required to implement the necessary process steps. Each saving is significant and if a sufficient number of savings can be made, the whole process can be carried out on a common serial processor.

The operational stages performed by apparatus for processing image data embodying the present invention is shown in Figure 11. Comparing the system of Figure 11 to the conventional system shown in Figure 2, 3-dimensional image data defining object space is read at the start and 2-dimensional video image data is generated at the end, in both systems. However, significant differences exist in the way in which the

data is processed, with an overall emphasis being placed on making economies in data transfer and data processing.

At step 111 object culling is performed by determining, after local transformation and viewing transformation, the extent to which an object can actually be seen. Extents are defined for the object in each dimension, that is to say, maximum and minimum values are determined for the object in each of the X, Y and Z dimensions. This process is facilitated by defining a cuboid bounding box of planes of constant X, Y or Z extent. It is not necessary to perform lighting upon the bounding box, therefore the local transformation and viewing transformation are performed as a common operation by concatanating the local and viewing matrices, thereby placing the object's bounding box within viewing space. Thereafter, the position of a transformed bounding box is considered, to determine whether it is totally within the viewable space, totally outside the viewable space or intersecting the viewable space.

If an object is totally outside the viewable space no further processing is performed on the polygons defining such an object, thereby making a significant

saving in processing time. For objects totally within the viewing space, processing continues on these polygons and no saving is made, however, the processing overhead for transforming the bounding box is quite modest. For objects which are partially within the viewable space and partially outside the viewable space, data defining the planes of intersection are stored for reconsideration at step 115, involving the clipping of polygons.

In the process shown in Figure 2, lighting calculations are performed for each iteration, irrespective of whether they are actually necessary. However, in the present embodiment a clock is present which effectively gives a unique time stamp to each iteration. This stamp is then applied each time lighting is moved and each time surfaces are moved. Thus, if on a present iteration, it can be seen that no changes have occured to the position or orientation of the lighting or to the position of polygons from the last interation, lighting calculations, calculated on the previous interation, may be used again, thereby making a significant saving in processing time.

At step 113 the lighting of polygons, where necessary, is performed. In the system shown in Figure 2, a local transformation is performed at step

21, surfaces are lit at step 22 and a view transformation is effected at step 23. As previously stated, the local transformation and the view transformation are mathematically similar and, when performed by matrix multiplication, the matrices defining the transformations may be concatanated into a common matrix. Thus, the local transformation and the viewing transformation are performed as one transformation from object space to viewing space, again significantly reducing the computational overhead.

Before the viewing transformation can be effected, it is necessary to perform lighting calculations. In the present embodiment, this is effected by calculating the inverse to the local transformation, effecting this inverse transformation upon the position of light sources, thereby effectively transforming the light sources into object space. Thereafter, lighting calculations are performed in object space and the object is transformed directly from object space to viewing space, using the concatanated matrix. Thus, the concatanated matrix used to transform the bounding box during the object cull, at step 111, is used again to transform the actual object polygons from object space into viewing

space at step 114.

Step 115 is substantially similar to step 24 in Figure 2 and involves the clipping of polygons which do not lie within the viewing space. However, as previously stated, additional information is generated at step 111 during the object culling process which identifies planes of viewing space through which objects intersect. This information facilitates the polygon clipping process at 115, thereby again reducing computational overhead to effect this part of the procedure.

The projection into a 2-dimensional image at step 116 is substantially similar to the projection process performed at step 25. At step 117 back face culling is performed, as required at step 26 of Figure 2 but this procedure is performed without reference to the polygon normals, which are only used in the present embodiment within object space and are not transformed into other spaces, again reducing computational overheads. Thus, other methods must be provided to perform back face culling at step 117, as described later.

Step 118 provides scan conversion and computational overhead for this process is reduced by taking account of the restricted size of most frame

buffers, typically only having eight bits per pixel location. Thereafter, a 2-dimensional video image is produced in similar fashion to the system shown in Figure 2 by raster scanning the frame buffer at video rate.

Aspects of the system shown in Figure 11 which differs significantly from the procedure carried out in the system of Figure 2 will now be described in detail.

OBJECT CULLING

Examples of the preferred embodiments in operation will be described with reference to an example similar to that described for the conventional system.

As previously described, an object is initially defined in its own object space, as shown in Figure 3. Data stored in memory actually consists of a table of vertices defined with reference to the X, Y and Z coordinates. The object culling process at step 111 consists of examining x, y and z coordinates so as to identify the extent of the vertices for an object in each dimension, that is to say the maximum and minimum value in each dimension is determined. From these extents, a bounding box is defined, as shown in Figure 12. Thus, plane 121 represents the maximum extent in the X direction and the plane opposite plane

121 represents the minimum extent in X. Similarly, plane 122 represents the maximum extent in Y and plane 123 represents the maximum extent in Z.

In the example previously described, the polygons in object space were instantiated three times into modelling space and the three instantiations were then transformed to viewing space. In the present embodiment, objects are transformed directly from object space to viewing space and such a transformation is carried out on the bounding box shown in Figure 12. Thus, when perspective views are being produced, the bounding boxs shown in Figure 12, are transformed into viewing space as shown in Figure 13.

In Figure 13, object 32b is completely within the viewable area. Object 32a is completely outside the viewable area and object 32c is partially inside and partially outside.

The polygons making up object 32b must be completely processed and no saving has been made. However, the polygons making up object 32a all exist within a bounding box which is outside the viewable space, therefore, on the particular interation under consideration, all these polygons are effectively rejected and no further processing is performed on

them.

The object 32c falls within a bounding box which is partially inside the viewing area and partially outside the viewing area. As far as this object is concerned, culling must be performed on a polygon basis. The fact that this object requires polygon culling is supplied to the polygon clipping process at step 115. Furthermore, information is also supplied to step 115 stating that the object intersects the X = + clip(X) plane. Thus, when doing comparisons at the polygon clipping process, only clipping to the positive X direction needs to be performed, such that polygons which have X coordinates greater than clip (X) are rejected while polygons having X coordinates less than or equal to clip (X) are retained.

Thus, by performing the transform on the points defining the bounding box, it is possible to determine the extent to which said points are transformed into the viewing space and thereby reject, totally, any objects which fall outside the viewing space. Furthermore, it is possible, to facilitate subsequent polygon clipping of objects which are partially within the viewing space.

The invention has been described with reference to a cuboidal bounding box, although any other bounding

shape may be used. The essential step is that of defining the extent of all points within the object.

TIME STAMPING TO AVOID LIGHTING CALCULATIONS

At step 112 changes are noted in light source parameters and changes are also noted in object positions. If a light source parameter and an objects orientation or position have not changed, it is not necessary to recalculate lighting characteristics for that particular object. Thus, lighting characteristics previously calculated are retained and used again.

Changes in lighting position, object position and object orientation are identified by applying a unique time stamp to each iteration of the process. Thus, if, in response to interactive operations, an object or a light source is modified, said objects or light source is restamped with the unique reference for that particular iteration. At step 112 the time stamps for these objects and lights are examined against their previous value for the previous iteration, thereby identifying a condition to the effect that a position or orientation has changed.

On identifying this condition, lighting characteristics are recalculated and no saving is made. However, if, say, one object is moved in a

space containing a plurality of objects, only the polygons making up that particular object require lighting characteristics to be recalculated and a significant saving can be made for the particular iteration.

LIGHTING POLYGONS IN OBJECT SPACE

a) Transforming Light Sources

In an interactive system, data is stored defining the position of light sources, a viewing position and an arrangement of polygons defining an object. In addition, a local transform defines how an object positioned in its own object space is transformed into viewable 3-dimensional space. This is usually referred to as modelling space and the system is arranged such that the viewer is given the illusion of moving around within this space.

In the present embodiment, the inverse of the local transform is calculated and this inverse transform is used to transform the position and/or orientation of light sources from modelling space and back into the objects own space. Thereafter, lighting characteristics of the non-transformed polygons, in their own object space, is determined in response to the transformed light source or light sources. Thereafter, the embodiment also transforms the object,

lit in its own object space, under an operation which combines the local transform and the viewing transform, thus a significant saving is made, in that two transformations have been replaced by a common transformation. Furthermore, as described with reference to Figure 5, it is necessary to consider the unit normal vector of each polygon when calculating lighting characteristics. In the present embodiment, the normal vector for each polygon is available within its own object space and the calculation using this unit vector is performed within object space. Thus, once lighting characteristics have been calculated, no further use of the unit normal vector is required and said vector is not transformed, thereby saving computational power.

A transformation of the unit normal vectors into modelling space usually results in said vectors no longer being of unit length. Consequently it would be necessary to renormalize said vectors, which represents a significant computational overhead. By performing the lighting calculations in object space, where the normal vectors are available, said normal vectors do not need to be transformed into modelling space for the purpose of calculating lighting. Thus, the overhead of performing this transformation is lost

along with the even greater overhead of performing the re-normalisation process.

Figure 14 shows a 2-dimensional representation of object space axes X and Y, with similar modelling space axes overlaid. A polygon 141 is shown, which represents the position of the polygon in its own object space. Data is stored representing the position of vertices in object space, along with data representing the polygon's unit normal vector 142.

Under the operation of a local transform, transforming the polygon 141 into modelling space, a polygon 143 is generated in modelling space. It should be noted that the local transform affecting this transformation consists of a translation in the positive X and Y directions, an enlargement by factor 2 and an anticlockwise rotation.

In modelling space, a light source 144 is present and the lighting characteristic of the polygon 143 would have been calculated with reference to the unit normal vector $\underline{N}$ and the unit light vector $\underline{L}$. The unit vector $\underline{N}$ would be obtained by tranforming unit vector 142, which produces a vector having a length greater than unity. Consequently, it would have been necessary to perform a process of re-normalization, in order to calculate the true normal unit vector $\underline{N}$.

In the present embodiment, lighting calculations are not performed with reference to a transformed polygon 143, therefore, there is no need to transform unit normal vector 142. As previously stated, the inverse transform to the local transform is calculated and this inverse transform is used to effect a transformation of the light source 144 to position 145 within object space. Thus, the relationship between the inversely transformed light source 145 and the non-transformed polygon 141 is equivalent to the relationship between the transformed polygon 143 and the non-transformed light source 144.

The unit vector $\underline{L}$ is calculated within object space and lighting characteristics are calculated from the dot product of $\underline{N}$ and $\underline{L}$. If required, other lighting characteristics such as specular lighting could also be calculated within object space. If required, inverse transformations could also be performed on the viewing position, thereby bringing the viewing position into object space. Thus, Phong model specular lighting could be calculated within object space. However, such lighting effects are not calculated according to the Phong model in the present embodiment, as detailed later.

Thus, as previously stated, significant

computational savings are made by effecting the inverse local transformation upon the light source rather than transforming the polygons into modelling space, so as to calculate lighting characteristics. The need to transform the unit normal vector into modelling space is avoided and a local transform and a viewing transform may be concatanated, so that only one operation is required to transform polygons from object space directly into viewing space.

b) Specular Highlights Listing

As described with reference to Figure 5, the calculation of lighting characteristics created by specular highlights is computationally demanding because values are usually raised to a power, so as to represent the way in which specular highlights rapidly fall off as movement is made from the position of highest intensity.

In the majority of applications, objects usually have few specular highlights, although such highlights add significantly to the quality of the final image. Some polygons may have many small specular reflections but very little impairment to the quality of the image results if such small insignificant highlights are ignored. Thus, for many objects, not all polygons have significant levels of specular

reflection. In most cases, the majority of polygons will not have significant levels of specular reflection. In the present embodiment, polygons which have significant levels of specular reflection are processed in accordance with a first algorithm and other polygons are processed in accordance with a second algorithm.

In the embodiment, the polygon data is arranged in a list so that polygons having a significant level of specular reflection are grouped together, such that all polygons requiring the first (expensive) algorithm are processed together and all polygons requiring the second (less expensive) algorithm can also processed together. By organising the polygons in this way, an algorithm or process for calculating specular reflections is only employed for polygons which have a significant amount of specular reflection thereon. As used herein, specular reflection relates to all types of reflection which requires a significant amount of processing of this type and thus may be separated from polygons which require only linear calculations for their lighting characteristics, predominantly diffuse lighting characteristics, to be calculated.

c) Highlight Value Calculation

The Phong model for calculating specular

reflection is detailed in Figure 5 and, as shown, it is necessary to calculate the dot product of the unit normal vector N with a unit vector H positioned half way between the vector L , directed towards the light source 41, and a vector E , directed towards the viewing position 42. Thus, implementing the Phong model is computationally expensive, given that the H vector must be calculated and the dot product of H with N must then be raised to a power.

In a preferred embodiment, lighting calculations are performed within object space, as previously described, therefore calculation of the H vector involves transforming the viewing position into object space and then performing additional calculations in order to determine the position of said vector.

Strictly speaking, specular highlights refer to highlights which are dependent upon the viewing position. Thus, in the Phong model, as the viewing position is changed, the position of the specular highlights also changes, the nature of such highlights lights being such that they have a relatively high intensity when the viewing position is lined up with the light source, which diminishes rapidly at other orientations.

Specular highlights significantly increase the

quality of the final image by giving it a sense of realism. However, as a result of experimentation, it has been established that it is the non-linear property of these highlights, the fact that they taper off very rapidly from a high intensity peak, that is the major factor in creating the effect. Furthermore, colours tend to de-saturate towards the colour of the light source. Thus, an observer is aware of the highlights as a feature of the object itself and not as a feature of the relationship between the viewing position and the light source.

The present embodiment makes use of the aforesaid experimental result. Lighting characteristics are calculated in response to lighting parameters of surfaces and the parameters of light sources, without making reference to the viewing position. Thus, a lighting parameter is non-linearly processed to simulate highlight characteristics, without reference to the view position.

In the embodiment, the $\underline{H}$ vector is not calculated and a specular highlight is simulated by considering the dot product of $\underline{N}$ with $\underline{L}$, as required for calculating diffuse characteristics.

To simulate the non-linear nature of such highlights, the specular characteristic is calculated by

raising the dot product of $\underline{N}$ with $\underline{L}$ to a power n, in which n represents the specularity or shinyness, of the polygon under consideration.

After calculating the non-linear term, said term is multiplied by the specular coefficient K(s) and a factor representing the intensity of the light source I(l).

As previously stated, in the embodiment, the specular calculation is only performed for polygons having a significant specular component. Thus, when K(s) falls below a particular threshold it is not necessary to perform the specular calculations.

d) Calculating Accuracy

Referring to Figure 4, illustrating objects 32a, 32b and 32c, light source 41 and a viewing position 42 in modelling space, the position of objects within this space are calculated to a high level of accuracey, using floating point arithmetic. A high level of accuracy is required because, given that many manipulations are performed, quantising errors would build up and could produce noticable effects in the output image. Furthermore, the use of floating point arithmetic allows objects to be scaled over very large ranges, such that a polygon forming part of an image may be viewed at one extreme (taking up substantially

the whole of the viewing field) and at another extreme, being hardly visible within a very small object.

In conventional systems, as previously described, all calculations are performed using floating point arithmetic of this type, including lighting calculations. However, although a very high level of accuracy may be provided for calculating lighting, most of this accuracy is truncated as data is supplied to the frame buffer 19, given that said frame buffer will have a limited depth, typically of eight bits per pixel location.

In the present embodiment, as previously stated, means are provided for calculating the position of objects in space in high accuracy floating point arithmetic. However, lighting characteristics are calculated using lower accuracy fixed point arithmetic. The limitations of fixed point arithmetic are tolerated because results are still produced which are accurate than the accuracy of values supplied to the frame buffer, therefore truncation is still required. However, the rate at which fixed point arithmetic may be performed is substantially higher than the rate of performing floating point arithmetic, thereby reducing computational demands.

In the embodiment, the fixed-point arithmetic is performed using 32 bits, of which 16 bits are allocated to whole numbers and 16 bits are allocated to fractions. The type of arithmetic employed is often referred to as integer arithemetic although, as will be appreciated, values perceived as integers by the arithmetic unit may actually represent fractions as previously described.

CULLING 2-DIMENSIONAL BACK FACES

In the conventional system, as previously described, back face culling is necessary once an image has been projected into 2-dimensions, that is to say, data representing points in 3-dimensions are converted to data representing the points in 2-dimensions.

In 2-dimensions, a plurality of polygons may be present at the same location. However, polygons which represent a back face are not actually visible and therefore said polygons must be culled from further processing.

In the conventional system, data representing the unit normal vector for each polygon also undergoes transformations. As previously stated, a transformation from object space into modelling space of unit normal vectors is necessary in the

conventional system because this vector is used for calculating lighting characteristics. However, in the present embodiment, given that lights are back transformed into object space, it is not necessary to transform the unit normal vectors, therefore these vectors are not availabe as data representing points in viewing space, nor are they available prior to projecting polygons into 2-dimensions.

In the present embodiment, it is still necessary to determine whether 2-dimensional polygons are forward facing or back facing, so as to effect culling of the back facing polygons. In the present embodiment, vectors defining the polygon edges are processed so as to determine whether a polygon is foward facing or back facing.

The vectors compared must share a common vertex, therefore a common vertex must be selected.

As described with reference to Figure 10, the process of scan conversion involves selecting the vertex 101 which has the minimum y coordinate, that is to say, it is the vertex which is closest to the top of the viewing screen. As part of the scanning process and as previously described, it is necessary to calculate the gradient of the vector connecting points 101 and 102, and to calculate the gradient of

the vector connection point 101 to 103. In the table of data, due to the anticlockwise convention, point 103 will be listed prior to point 101, which in turn will be listed prior to point 102. Thus, for a foward facing polygon, irrespective of its transformation within 3-dimensional space, the vector connecting point 101 to point 102 should always be to the left of the vector connection point 101 to point 103, when the polygon is front facing.

Thus, the gradient, that is to say the rate at which X changes with Y, will always be larger for the vector connecting vertex 101 to vertex 103 than the similar gradient for the vector connecting vertex 101 to vertex 102. Thus, if the gradient of the vector connecting the highest (minimum y value) vertex to its predecessing vertex is greater than or equal to the gradient connecting said highest vertex to its succeeding vertex, the polygon is to be culled. This is because it is either back facing or it is on edge, neither of which should be rendered.

As an alternative to comparing gradients, in an alternative embodiment, the cross product of vectors connecting points 103, 101 and 102 is calculated. Such an approach is similar to transforming the normal vectors, as previously performed. However, the

magnitude of the vector is irrelevant and only its direction is required, thereby making the calculation less intense than that required for re-normalization.

Thus, a cross product is calculated to identify a normal vector and the polygon is culled if the vector calculated in this way points into the screen.

SCAN CONVERSION

Referring to Figure 11, at step 116 the 3-dimensional view is projected to define an image in 2-dimensions, wherein each vertex is given a 2-dimensional coordinate position. At step 117 back face culling is performed as previously described so that, at any position on the screen, only one polygon is present, thereby uniquely defining the characteristics of that position.

Step 118 consists of scan conversion, which involves specifying which pixels are required to be modified to define a particular polygon and, additionally, to specify a colour for thos particular pixels.

In the preferred embodiments, the process for identifying which particular pixels are to be included within a polygon is substantially similar to the process employed in the conventional system and as

described with reference to Figure 10. The process is also described in the applicants co-pending European patent application no. 9119141.1, included herein as part of the present disclosure.

As previously described, it is necessary to determine gradients, specifying how X varies, that is the span across each line, as Y varies on a scan line by scan line basis. Thus, gradients, conventionally referred to as variations of Y with respect to X are, in accordance with the present convention, referring to variations in X with respect to Y. Thus, as previously stated, as Y is incremented, identifying descending scan lines, it is possible to identify pixel locations within the polygon which require colour modification.

In the preferred embodiments, each pixel location within the frame buffer 19 stores eight bits and, as previously stated, in the preferred embodiments, lighting characteristics are calculated using fixed point arithmetic, although these fixed point values will still require further truncation before being written to the frame buffer 19. In the embodiments, 2-dimensional image data is produced, wherein pixels in 2-dimensional areas are coloured in response to calculated lighting characteristics, representing the

colour of said areas. Pixel colours are stored in a look-up table, which is addressed by values read from the frame buffer. Values stored in the frame buffer have a predetermined number of bits representing colour-hue and a predetermined number of bits representing another characteristic of the selected colour.

In a preferred embodiment, the eight bit addresses to the look-up table, sequentially read from the frame buffer, are arranged such that three bits identify hue and the remaining five bits represent another characteristic of the selected hue. Thus, with three bits available, a total of eight hues may be provided, selected from the gamut of colours which may be realised by the VDU 18. Thus, for example, hues such as blue, red, green magenta, cyan and yellow may be selected and the system may be considered as having a separate five bit look up table for each of said selectable hues.

In a preferred embodiment, within each of said five bit look-up tables, the other characteristic which varies is luminance, which varies linearly. This arrangement is shown graphically in Figure 16. Three bits of a value read from the look up table represent colour, therefore one of the colours C1 to

C8 is selected.

The remaining five bits define one of 32 luminance levels. Thus, linear interpolation between colours is not possible and colour aliasing occurs. However, luminance is controllable over a 32 bit range, which produces results of acceptable quality, given that the eye is more sensitive to luminance aliasing.

A colour ramp for an alternative embodiment is shown in Figure 17. It should be noted that this colour ramp is repeated eight times, in a similar fashion to Figure 16, once for each selectable colour.

In Figure 17, luminance is varied in response to the additional five bits but the variation of luminance with values read from the frame buffer 19 is non-linear to provide gamma correction, thereby allowing look-up tables for colour conversion and gamma correction to be combined into a common look-up table.

An alternative colour ramp is shown in Figure 18 which again would be repeated eight times as shown in Figure 16. In this alternative embodiment, the other characteristic of a selected colour is saturation. Thus the colour ramp may initiate at a fully saturated colour value, with a predetermined amount of luminance

and, as input values increase, the saturation decreases, so that the output colour fades from a distinct colour towards white.

The purpose of the colour mapping shown in Figure 18 is to simulate the effect of specular highlights which appear substantially white, with white light sources, even on coloured objects. Again, this response is made non-linear, which facilitates the calculation of specular highlight non-linearities. Thus, as previously described, specular highlights may be calculated by raising the dot product of N and L to a power. In the present alternative embodiment, it may only be necessary to calculate the dot product without raising to to power, because the non-linear raising to a power may be performed by the look-up table, when pixel values stored in the frame buffer are converted to real colour signals.

Alternatively, a specular highlight process may perform easily implemented power raising (to the power 16 say) with further non-linearity being introduced by the look-up table.

An alternative embodiment for the colour ramps is shown in Figure 19, which combines the luminance characteristic of Figure 17 with the saturation characteristic of Figure 18. As previously described,

three bits select one of the eight available hues. The remaining five bits represent the level of lighting applied to the surface. Initially, luminance values start at zero and the object is represented as black. As the lighting value increases, the luminance value increases and saturation remains high, such that a red area, say, gets brighter but remains red, representing the effect of diffuse reflection, until at about half way, luminance reaches its maximum value.

As the lighting level increases, it is assumed that this increase is due to specular reflection. Thus high input values result in a decrease in saturation, so that a red highlight changes to white. Thus, each ramp undergoes a change from black to full colour to white, wherein a different colour is provided for each ramp.

As shown, the luminance region varies non-linearly to take account of gamma correction and the saturation region varies non-linearly to represent specular reflection. Alternatively, either or both of these ramps could be linear.

Many features of the preferred embodiments reduce the computational demands for a 3-dimensional interactive graphics engine. When taken together,

they allow interactive 3-dimensional graphics to be generated using a common, programable central processing unit, with effectively no requirement for purpose built hardware. However, it should also be understood that many sub-combinations may be selected, so as to reduce the amount of purpose built hardware required, thereby providing significant benefit.

The techniques disclosed herein are particularly suitable to interactive graphics environments. However, the techniques may also be employed in other graphics environments, such as non-interactive environments or real-time environments, where the speed of interactivity is perceived to be instantaneous.

CLAIMS

1. Apparatus for processing image data representing multi-dimensional objects, said appartus including processing means and data storage means, wherein said storage means stores data defining;

(a) the position of a light source,

(b) a viewing position,

(c) an arrangement of polygons defining an object and

(d) a local transform for transforming said object into viewable 3-dimensional space,

characterised by

said processing means including means for:

(e) defining the extent of points for an object in each dimension;

(f) performing said local transform on said points and

(g) determining the extent to which said points are transformed into viewing space.

2. Apparatus according to claim 1, wherein said points define a cuboidal bounding box.

3. Apparatus for processing image data according to claim 1, wherein said viewable space has a shape which takes account of a perspective view.

4. A method of processing image data representing multi-dimensional objects, comprising the steps of storing data relating to:

(a) the position of a light source,

(b) a viewing position,

(c) an arrangement of polygons defining an object and

(d) a local transform for transforming said object into viewable 3-dimensional space,

characterised by

(e) defining the extent of points for an object in each dimension;

(f) performing said local transform on said points and

(g) determining the extent to which said points are transformed into viewing space.

5. A method according to claim 4, wherein said points define a cuboidal bounding box.

6. A method according to claim 4, wherein said viewable space has a shape which takes account of perspective.

Patents Act 1977
Examiner's report to the C mptroller under Secti n 17 (The Search Rep rt)

Application number
GB 9220792.7

Relevant Technical fields

(i) UK Cl (Edition K) H4T (TBBA, TBEA, TCGA, TCGD, TCGG, TCGK, TCGX, TCJA, TCXX)

(ii) Int Cl (Edition 5) G06F

Search Examiner
KING

Databases (see over)
(i) UK Patent Office

(ii)

Date of Search
30 DECEMBER 1992

Documents considered relevant following a search in respect of claims 1 TO 6

Category (see over)	Identity of document and relevant passages	Relevant to claim(s)
X	GB 2194715 A (IBM) see abstract	1, 4
X	GB 2194656 A (IBM) see abstract	1, 4
X	EP 0152741 A2 (PHOENIX DATA) see abstract	1, 4

~~SJJ - doc99\fil001235~~

Category	Identity of document and relevant passages	Relevant to claim(s)

-56-

Categories of documents

X: Document indicating lack of novelty or of inventive step.

Y: Document indicating lack of inventive step if combined with one or more other documents of the same category.

A: Document indicating technological background and/or state of the art.

P: Document published on or after the declared priority date but before the filing date of the present application.

E: Patent document published on or after, but with priority date earlier than, the filing date of the present application.

&: Member of the same patent family, corresponding document.

Databases: The UK Patent Office database comprises classified collections of GB, EP, WO and US patent specifications as outlined periodically in the Official Journal (Patents). The on-line databases considered for search are also listed periodically in the Official Journal (Patents).